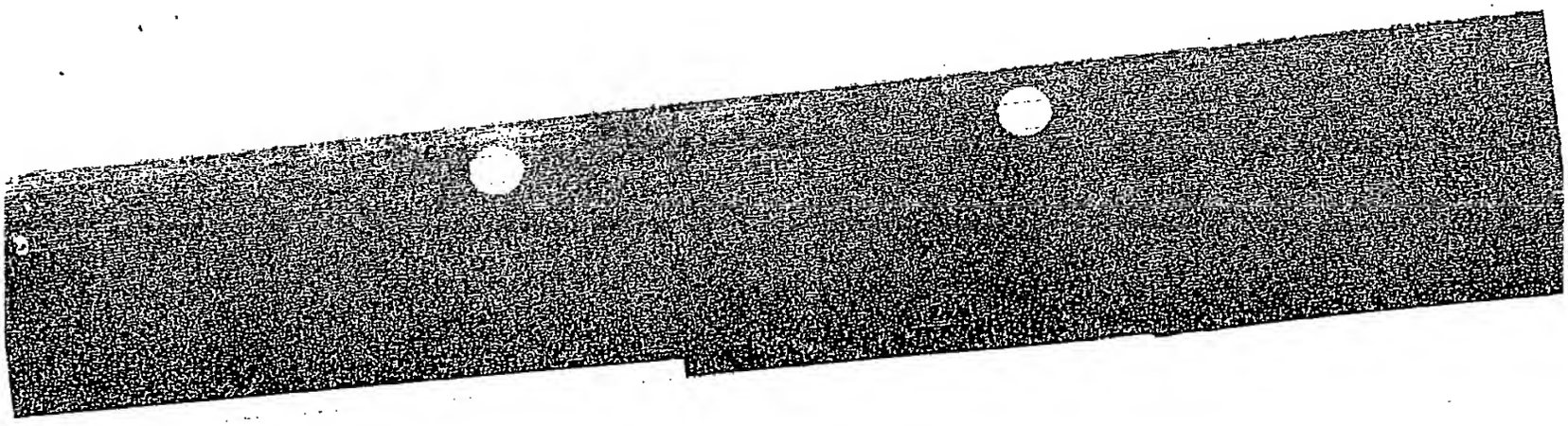
Appendix I

DICTIONARY OF COMPUTING

THIRD EDITION

Oxford New York Tokyo
OXFORD UNIVERSITY PRESS
1990



Oxford University Press, Walton Street, Oxford OX2 6DP
Oxford New York Toronto
Delhi Bombay Calcutta Madras Karachi
Petaling Jaya Singapore Hong Kong Tokyo
Nairobi Dar es Salaam Cape Town
Melbourne Auckland
and associated companies in
Berlin Ibadan

Oxford is a trademark of Oxford University Press

Published in the United States
by Oxford University Press, New York

First published 1983
Reprinted 1983, 1984, 1985
Second edition 1986
Third edition 1990

British Library Cataloguing in Publication Data
Dictionary of computing. – 3rd ed. –
(Oxford science publications)
1. Computer systems
004
ISBN 0 19 853825 1

Text prepared for automatic typesetting by
Market House Books Ltd, Aylesbury
Printed and bound in Great Britain by
Courier International Ltd
Tiptree, Essex

The great advances in the
theory, technology, and a
growth in the uses to wh
people using them. As th
ing terminology. For thi
over 550 new entries ha
have been extensively up
of computing, especially
puter organization and
and developments in thi
puting, networking, an
a single alphabetical lis
associated fields of ele
computing covered in

algorithms and th
programming lan
program develop
data structures a
operating system
computer organi
hardware, inclu
computer comm
information tec
computer appli
major compute
legal aspects o

The entries in the
branches of comp
range from basic
level computer sc
tables. The dictio
science and of al
be a valuable ref
of computing as

A major unde
the dictionary h
Market House
and appreciatio
effort.

September 198

databank A system that offers facilities for the deposit and withdrawal of data to a community of users on a particular topic (e.g. biological species, trade statistics, commodity prices). While it need not be an open public facility, the usual implication is that the user community is widespread. Access to a databank may be, for instance, via a *videotex facility, or via any other form of *network, or even via the postal service. The data itself may be organized as a *database or as one or more *files.

database 1. Normally and strictly, a *data file that is defined and accessed using the facilities of a *database management system (DBMS); this implies in particular (a) that it is defined by means of a *schema that is independent of any programs that access the database (*see* data independence), (b) that is uses *direct access storage.

The use of a DBMS offers better control over data (*see* database administrator), and more sophisticated file organization and access methods, than are otherwise normally available. Because of the latter point, databases can be more complex than conventional files, and they often integrate the data previously held in many separate files. Size and complexity, however, are not necessary characteristics of databases: the availability of DBMS software on microcomputers, for instance, leads to many small and simple databases, where the advantage gained by using a DMBS is ease of definition and access.

Database use may be partitioned into three roles: end-users (who supply and/or receive the data), programmers (who write *applications programs to process the data), and *database administrators. For a large database, each role may be performed by many people; for a small database, one individual may carry out all three roles.

See also database language, database system, data dictionary.

2. Occasionally and loosely, one or more *data files, however defined, accessed, and stored, that hold nontransient data in a computer application.

database administrator (DBA) A person responsible for the specification, design, implementation, efficient operation, and maintenance of a *database. The identification of a distinct role for database administration follows from the concept of *data independence, and from the realization that databases form an important and valuable corporate resource. The DBA works with users in establishing database requirements, as part of the activity of system specification; he uses a *data description language for database definition, as part of the activity of system design; he works with programmers whose programs need to access the database; he is responsible for loading the database with data, as part of the activity of system implementation; he monitors the performance of the database, using available hardware/software tools, to determine when the data should be reorganized or the database redesigned.

Overall, the DBA should seek to achieve *database integrity, *security, and efficiency. He is particularly concerned with balancing the conflicting requirements (of end-users and programmers) arising from the fact that the database may be shared by a number of different applications. Two distinct roles are emerging: a business-modeling role, and a technical role of making the database management system work.

database integrity The condition of a database in which all data values are correct, in the sense (a) of reflecting the state of the real world – within given constraints of accuracy and timeliness – and (b) of obeying rules of mutual consistency. The maintenance of database integrity involves integrity checking, and recovery from any incorrect state that may be detected; this is the responsibil-

ity of a *database administrato
the facilities of a *database man
system.

File integrity can be defined i
terms. Typically, however, files
ject to less extensive integrity
than databases.

database language A generic t
ring to a class of languages
defining and accessing *da
database language compris
more *data description lan
one or more *data manip
guages; these may be referre
sublanguages. An importan
is whether a database
designed (a) to be used as
to an existing programm
(which is referred to as
guage), or (b) to be used
of any programming lang
case it may be referred to
ing), or (c) to be used in
particular database lan
associated with a partic
management system. Bo
with database language
include *ANSI/SPARC
and *ISO.

database management sy
software system with
*database language pro
both (b) the handling
for database access
programs and/or end
maintenance of *dat
DBMS / thus has fe
both with *compilers
ing systems, and ma
the level of *abstrac
tems offer to prog
users.

Three classes of
tionally been disti
*hierarchical, *net
database systems
between these c
harder to maint
seems likely tha

sionally and loosely, one or
ata files, however defined,
and stored, that hold nontran-
in a computer application.

administrator (DBA) A person
e for the specification, design,
tation, efficient operation, and
ıce of a *database. The identi-
f a distinct role for database
ation follows from the concept
independence, and from the
ı that databases form an
and valuable corporate
The DBA works with users in
ng database requirements, as
he activity of system specifica-
uses a *data description lan-
database definition, as part of
ity of system design; he works
grammers whose programs need
the database; he is responsible
ng the database with data, as
he activity of system implemen-
e monitors the performance of
oase, using available hardware/
tools, to determine when the
uld be reorganized or the data-
esigned.
l, the DBA should seek to
*database integrity, *security,
iency. He is particularly con-
with balancing the conflicting
ents (of end-users and program-
rising from the fact that the
may be shared by a number of
applications. Two distinct roles
ging: a business-modeling role,
chnical role of making the data-
nagement system work.

integrity The condition of a
in which all data values are
in the sense (a) of reflecting the
the real world – within given
its of accuracy and timeliness –
of obeying rules of mutual con-
The maintenance of database
involves integrity checking, and
from any incorrect state that
detected; this is the responsibil-
ity of a *database administrator, using the facilities of a *database management system.

File integrity can be defined in similar terms. Typically, however, files are subject to less extensive integrity checking than databases.

database language A generic term referring to a class of languages used for defining and accessing *databases. A database language comprises one or more *data description languages and one or more *data manipulation languages; these may be referred to as *data sublanguages*. An important distinction is whether a database language is designed (a) to be used as an extension to an existing programming language (which is referred to as the *host language*), or (b) to be used independently of any programming language (in which case it may be referred to as *free-standing*), or (c) to be used in either way. A particular database language will be associated with a particular *database management system. Bodies concerned with database language standardization include *ANSI/SPARC, *CODASYL, and *ISO.

database management system (DBMS) A software system with facilities for (a) *database language processing, to permit both (b) the handling of run-time calls for database access from application programs and/or end-users and (c) the maintenance of *database integrity. A DBMS thus has features in common both with *compilers and with *operating systems, and may be seen as raising the level of *abstraction that those systems offer to programmers and end-users.

Three classes of DBMS have conventionally been distinguished, supporting *hierarchical, *network, and *relational database systems. The distinctions between these classes are becoming harder to maintain, however, and it seems likely that other classes will emerge; this conventional classification is thus losing its usefulness.

The following are well-known DBMSs: ADABAS, *dBASE IV, DBOMP, DMS-1100, IDMS, IDS, *IMS, *INGRES, MDBS, *NOMAD, QBE, TDMS, and TOTAL.

database recovery The process of restoring *database integrity once a database has been found to be incorrect. *See also* recovery log.

database system 1. A *database together with its *database management system (software) and storage devices (hardware). A database system is a complex form of *associative memory. A data item in a database is typically associated with many other items, some of which may be physically contiguous (i.e. in the same *record) and others not. Data is usually retrieved by giving values of specified items, in order that the system should respond with the values of specified associated items. For example, a system might retrieve employee name and annual salary for a given employee number, or the registration numbers of all cars of a given color, make, and year of registration.

2. *Short for* database management system.

data break (cycle stealing) *See* direct memory access.

data bus (data path) A group of signal lines used to transmit data in parallel from one element of a computer to another. The number of lines in the group is the *width* of the data bus, each line being capable of transferring one bit of information. In a mainframe the width of the data bus is typically equal to the word length, i.e. 32, 48, or 64 bits. The data bus used to interconnect LSI components need not have the same width as is used on the chips themselves. For example, a processor with an internal data bus width of 32 bits could

procedure for producing the result takes the form of a ce of operations, and thus with tive languages the notions of flow itrol and ordering of statements herent. Such a language is typi- characterized by the presence of ssignment statement, which, being ctive (the assigned value replaces evious value of the variable), also ds on the notion of ordering. ative languages are closely associ- with the von Neumann model of itation, and the majority of widely languages – including COBOL, RAN, Algol, and Pascal – are ative.

nentation The activity of proceed- om a given design of a system to a ng version (known also as an *nentation*) of that system, or the ic way in which some part of a n is made to fulfill its function. example, a control unit may be mented by random logic or by programming; a multiplier may be mented by successive additions and or by a table look-up. Another ple occurs in computer families, different implementations may in the type of circuit elements or in the actual parallelism (as sed to logical parallelism) of the

th software, use of the term nor- implies that all major design deci- have been made so that the imple- ation activity could be relatively ghtforward. For many systems a ber of important characteristics may become bound until the implemen- n activity; examples include the ramming language in which the sys- is written, the type of computer loyed, the actual hardware configur- n, or the operating system used. such systems there may be a ber of distinct implementation activ- in order to provide several versions he system, e.g. written in different languages or operating on different hardware.

implicant A *product term that covers at least one of the *standard sum of product terms in a *Boolean function, but will introduce no new (unwanted) standard sum of product terms.

A *prime implicant* is an implicant that includes a *standard product of a function that is not otherwise included.

implied addressing (inherent addressing) A type of *addressing scheme, the term referring to the fact that in many instruction formats the location of one or more operands is implied in the instruction name and is specified in the instruction description. An implied address is usually that of one of the machine registers.

import list In modular languages such as *Modula-2, a list of the names used inside a module that are declared in other modules.

impulse noise *Noise of large amplitude and some statistical irregularity, affecting an analog channel severely but (relatively) infrequently. In contrast, *white noise affects it (relatively) unseverely but continuously. Impulse noise affecting an analog channel carrying a binary signal usually causes *burst errors.

IMS *Trademark* A well-established *database management system provided by IBM. It was originally a *hierarchical system but has gained additional nonhierarchical features as a result of practical needs.

inactive Not running (pertaining to the state of a process).

incidence matrix A representation of a *graph G employing a *matrix in which there is a row for each vertex v of G. The entries on this row are just the vertices that are joined by an edge to v.

inclusive-OR gate *Another name for* OR gate.

inclusive-OR operation *Another name for* OR operation, making explicit the difference between this and the *exclusive-OR operation.

incompleteness theorems *See* Gödel's incompleteness theorems.

incremental compiler A compiler that can compile partial programs, and can compile additional statements for a program without recompiling the whole program. Incremental compilers were at one time in vogue for interactive programming, but interactive language systems nowadays are almost always implemented in an interpretive manner.

incremental plotter A device that can draw graphs and other line images when fed with digital data. The plotter forms the image by moving a pen or the paper or both in a succession of increments. The increments are typically 200 per inch for drum plotters and can be 500 per inch for flatbed plotters. *See also* plotter.

indegree *See* degree.

indeterminate system A logic system whose *logic states are unpredictable.

index 1. A set of *links that can be used to locate records in a *data file. In a *single-level index*, and at the lowest level of a *multilevel index*, an entry points directly to an individual record or to a group of records. At the higher levels of a multilevel index an entry points to a group of entries at the next lower level; a multilevel index is used where the size of a file would give rise to excessive search time using a single-level index. The B+ tree (*see* B-tree) is an efficient form of multilevel index. *See also* indexed file.

2. The value held in an *index register.

3. (subscript) An integer value selecting a particular element of an *array. In some high-level languages, arrays can be indexed by ordered sets of discrete values that are not integers, such as the names of the days of the week or the names of the chemical elements.

indexed addressing (indexing) A method of generating an *effective address that modifies the specified address given in the instruction by the contents of a specified *index register. The modification is usually that of addition of the contents of the index register to the specified address. The automatic modification of index-register contents results in an orderly progression of effective addresses being generated on successive executions of the instruction containing the reference to the index register. This progression is terminated when the index register reaches a value that has been specified in an index-register handling instruction.

indexed file A *data file in which *records can be accessed by means of an *index. If the same field is used both in the index and for sequencing the records in the file, the index is called a *primary index* (and the file is called an *indexed sequential file*). Otherwise the index is called a *secondary index* (*see* inverted file).

indexed sequential file A file combining properties of *random-access files and *sequential files. *See* indexed file, ISAM.

indexing *See* indexed addressing.

index register A register that can be specified by instructions that use *indexed addressing. An index register is usually controlled by one or more instructions with the ability to increment or decrement the register by a fixed amount, to test the register for equality with a specified value (often zero), and to jump to a specified location when equality is achieved. It is part of the *processor status word.

indicator 1. A bit or bit configuration that may be inspected to determine a status or condition. Examples are an *overflow bit, a device status, any portion of the *program status word. *See also* qualifier register.

2. A visual, sometimes aural, indication of the occurrence of a specific status or condition, e.g. system running (halted), undefined instruction.

indirect addressing A method of addressing in which the contents of the address specified in the instruction (which may itself be an *effective address) are themselves an address to be used to provide the desired memory reference. Two memory references are thus needed to obtain the data.

One use of indirect addressing is to supply a way of circumventing short address field limitations since the first memory reference provides a full word of address size. Another use is as a pointer to a table. Since an operand is not available at the usual time in the fetch-execute cycle, completion of that cycle must be deferred until the operand is finally available. Indirect addressing is thus sometimes referred to as *deferred addressing*.

induction A process for proving mathematical statements involving members of an ordered set (possibly infinite). There are various formulations of the principle of induction. For example, by the *principle of finite induction*, to prove a statement $P(i)$ is true for all integers $i \geqslant i_0$, it suffices to prove that

(a) $P(i_0)$ is true;

(b) for all $k \geqslant i_0$, the assumption that $P(k)$ is true (the *induction hypothesis*) implies the truth of $P(k+1)$.

(a) is called the *basis* of the proof, (b) is the *induction step*.

Generalizations are possible. Other forms of induction permit the induction

step to assume the truth of
also that of
$P(k-1), P(k-2), \ldots, P$
for suitable i. Statements of s
ables can also be considere
structural induction.

inequality A *binary relatio
cally expresses the relative n
two quantities, usually num
more generally elements of
ordered set (*see* partial orde

The inequalities defined
gers usually include

$<$ (less than)
$\leqslant$ (less than or equal to
$>$ (greater than)
$\geqslant$ (greater than or equa
$\neq$ (not equal to)

A similar set of inequalit
defined on the real numl
equalities can produce err
in programming languag
the inherent inaccuracies i
numbers are usually re
floating-point notation).

The term inequality is
to any comparison invc
expressions and using t
bols. A special case is
equality:

$$|a + b| \leqslant |a|$$

where $|\ |$ denotes the
function.

inference The formal m
ing that underlies lo
Rules of inference pr
whereby a logician or
program (*see* theorem
previous results, theore
derive new results. S
tional calculus, predica

inference engine With
*expert systems, the
system program that
*knowledge base an
ences. If the knowled
as a program then t
is the interpreter. The